

Software Construction and Development

assignment 1



Submitted On: October 27, 2025

Developed By:

Muhammad Shamoil (232025)

Muhammad Zeeshan (232035)



Contents

1.0 Introduction.....	3
1.1 Purpose	3
1.2 Scope.....	3
1.3 Limitations.....	3
2.0 System Requirements.....	4
2.1 Functional Requirements	4
2.2 Non-Functional Requirements	5
3.0 Actors and Roles.....	6
4.0 UML Diagrams.....	7
Class Diagram.....	7
4.2 Use Case Diagram.....	7
4.3 Activity Diagram.....	8
4.3.1 Login	8
4.3.2 Job Application.....	8
4.3.3 Job Posting.....	9
4.4 Sequence Diagram	9
4.4.1 Job Application.....	9
4.4.2 Job Posting.....	9
5.0 Glossary.....	10
6.0 References	10

Software Requirement Specification Document

GET A JOB (WEB PORTAL)

1.0 Introduction

1.1 Purpose

This document specifies the software requirements for the "GetAJob" web application. It outlines the system's goals, features, and limitations, serving as a foundational agreement for project development. The "GetAJob" system is a web-based job portal built on Java, Tomcat, and an MVC architecture.

1.2 Scope

The system will provide a platform for Job Seekers to manage their professional profiles and apply for jobs, and for Job Posters (Employers) to manage job listings and view applicants. It will handle user authentication, profile management, job posting, and the application process.

1.3 Limitations

- This initial version will **not** include an administrator back-end.
- This version will **not** include an automated job recommendation engine.
- This version will **not** include an email or real-time notification system.

2.0 System Requirements

2.1 Functional Requirements

2.1.1 User Management

1. Users must be able to register as either a Job Seeker or a Job Poster.
2. The system must validate user credentials during registration and login.
3. Authenticated users must be able to edit, update, and delete their own accounts.
4. All user passwords must be encrypted and stored securely in the database.

2.1.2 Job Seeker Profile

5. Job seekers must be able to create a personal profile with details like name, education, skills, and experience.
6. Job seekers must be able to upload a CV/resume.
7. Job seekers must be able to update or delete their profiles.
8. The system must allow job seekers to set job preferences (e.g., location, job type, salary).

2.1.3 Job Posting

9. Employers (Job Posters) must be able to create, edit, and delete job posts.
10. Job posts must include fields for title, description, location, salary, required skills, and application deadline.
11. Employers must be able to view a list of all jobs they have posted and the applicants for each job.

2.1.4 Job Application

12. Job seekers must be able to apply for available jobs.
13. The system must prevent duplicate applications by the same user for the same job.
14. Job seekers must be able to view a list of all jobs they have applied for and the status of each application.
15. Employers must be able to accept or reject applications.

2.1.5 Search and Filter

16. All users (registered or guest) must be able to search for jobs by keywords, titles, or skills.
17. All users must be able to filter search results by category, location, or salary.

2.2 Non-Functional Requirements

1. **Performance:** The system must respond within 2 seconds under normal load. All database queries must be optimized.
2. **Scalability:** The system must support horizontal scaling. The architecture should be modular to allow for future ML integration (e.g., recommendation engine).
3. **Security:** User sessions must expire after a defined period of inactivity.
4. **Maintainability:** The system must use an MVC (Model-View-Controller) architecture for separation of concerns. System configurations must be externalized in XML files.
5. **Reliability and Availability:** The system must have an uptime of 99.5%.
6. **Usability:** The user interface (UI) must be intuitive, responsive (mobile-friendly), and consistent across the application.
7. **Portability:** The application must run on any platform that supports Java 11+ and Apache Tomcat. The database layer must support MySQL.
8. **Data Integrity:** All primary records (users, jobs) must use unique identifiers. Database transactions must follow ACID properties.
9. **Logging:** All critical user actions and system errors must be logged securely with timestamps.

3.0 Actors and Roles

The system defines three primary actors who interact with the system.

1. User (Base Actor):

This is a general role for any person interacting with the system. They inherit base-level permissions.

Responsibilities:

Can browse and search for jobs. To perform any other action, they must register and assume one of the roles below.

2. Job Seeker:

An authenticated User whose goal is to find employment.

Responsibilities:

1. Manages their personal profile (create, update, delete).
2. Uploads and maintains their resume.
3. Searches and filters job listings.
4. Applies for jobs.
5. Views the status of their submitted applications.

3. Job Poster (Employer):

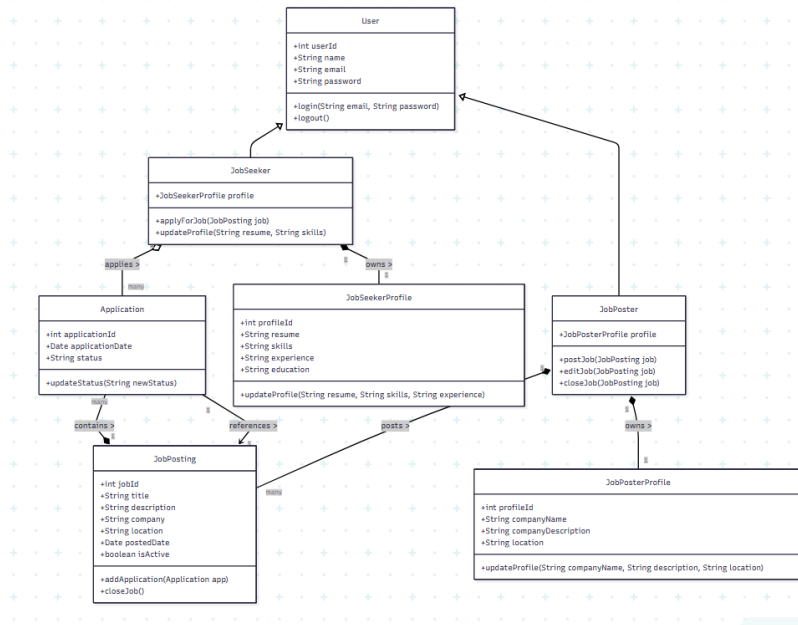
An authenticated User whose goal is to find candidates for a job.

Responsibilities:

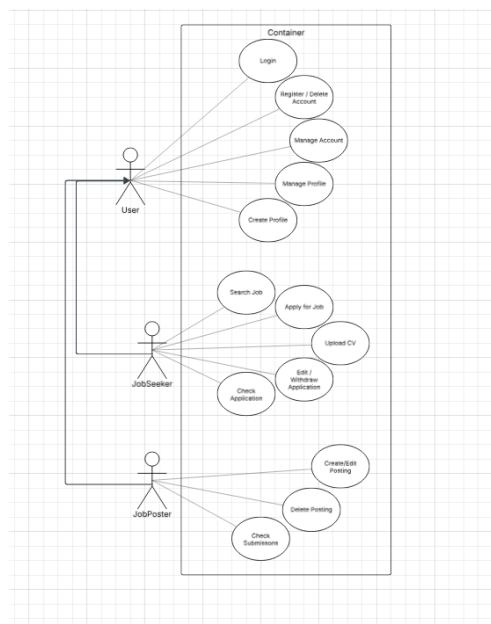
1. Manages their company profile.
2. Posts, edits, and deletes job listings.
3. Views all applicants for their specific job postings.
4. Updates the status of an application (e.g., "Accepted", "Rejected").

4.0 UML Diagrams

Class Diagram

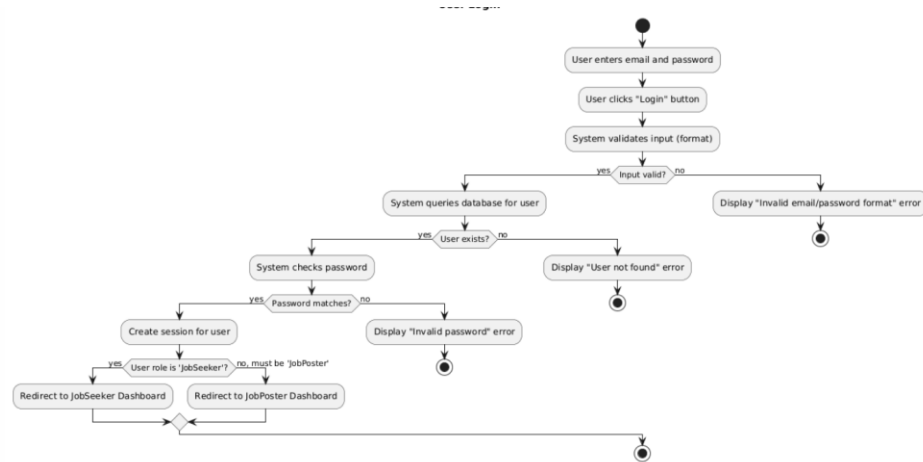


4.2 Use Case Diagram

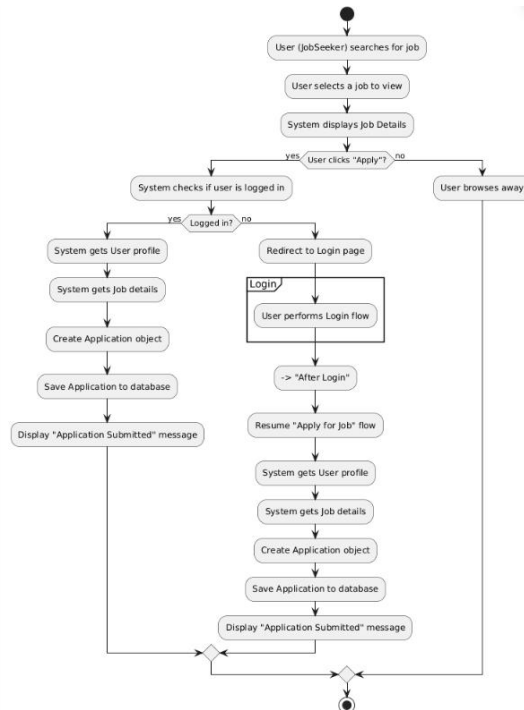


4.3 Activity Diagram

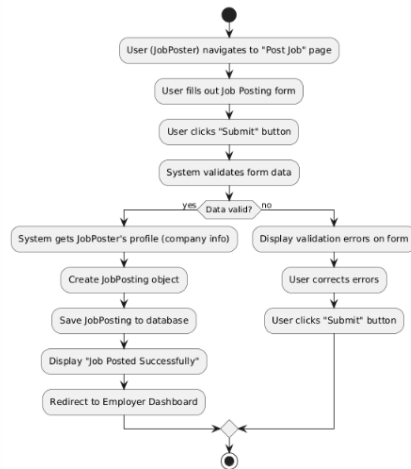
4.3.1 Login



4.3.2 Job Application

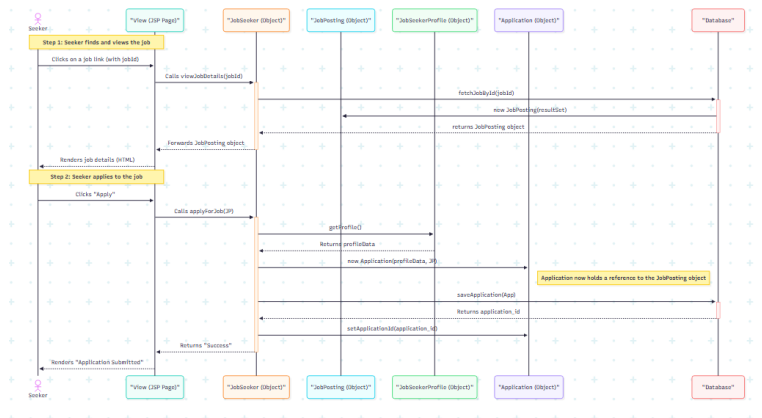


4.3.3 Job Posting

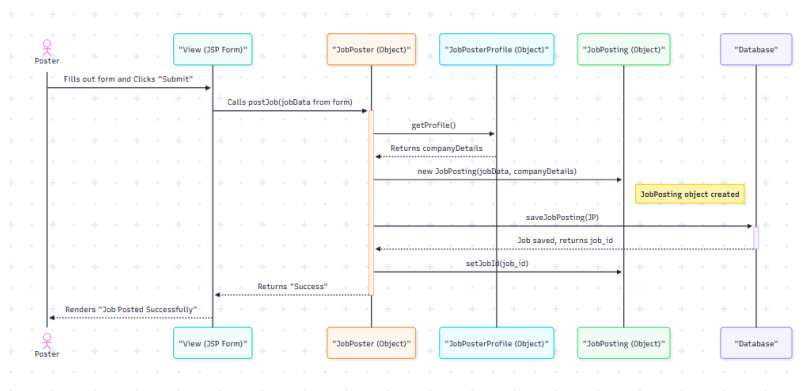


4.4 Sequence Diagram

4.4.1 Job Application



4.4.2 Job Posting



5.0 Glossary

1. ACID: Refers to Atomicity, Consistency, Isolation, and Durability. These are properties of database transactions to guarantee validity.
2. Java: The core programming language used to build the "GetAJob" web application.
3. JSP (JavaServer Pages): A technology used to create the dynamic web pages (the "View") that the user interacts with, as shown in the sequence diagrams.
4. ML (Machine Learning): Mentioned as a potential future enhancement for features like a job recommendation engine.
5. MVC (Model-View-Controller): The software architectural pattern used to organize the project, ensuring separation of concerns.
6. MySQL: The specific database technology the system is required to support.
7. SRS (Software Requirement Specification): This document, which outlines the goals, features, and requirements of the "GetAJob" portal.
8. Tomcat (Apache Tomcat): The web server and servlet container required to run the Java application.
9. UI (User Interface): The visual layout and design of the application that the user interacts with.
10. UML (Unified Modeling Language): The standardized modeling language used to create the diagrams in Section 4.0.
11. XML (Extensible Markup Language): The file format used to store external system configurations.

6.0 References

N/A. No external documents, standards, or publications were cited in the creation of this specification.